

Apuntes de Algoritmos Genéticos

Parte I: Introducción

M. Angélica Pinninghoff J. - Ricardo Contreras A.
Magíster en Inteligencia Artificial
Universidad Adolfo Ibáñez

2022

1 Preliminares

Si bien es cierto los algoritmos genéticos son mecanismos para buscar soluciones, típicamente en el área de optimización, su inspiración es biológica.

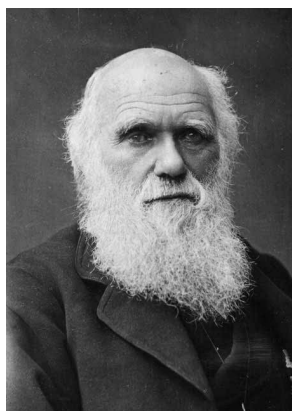


Figure 1: Charles Darwin. Autor de la Teoría de la Evolución.

La evolución biológica es un proceso de cambio continuo que lleva a la diversificación de las especies a través del tiempo. Los cambios evolutivos afectan a los seres vivos en su morfología, fisiología y comportamiento, entre otros aspectos.

La evolución es una propiedad inherente a los seres vivos. La idea de que la vida ha evolucionado a partir de un ancestro común es algo que está en la discusión científica desde la época de los filósofos griegos. Charles Darwin y Alfred Russel Wallace propusieron de manera independiente, en 1858, que la selección natural es el mecanismo fundamental que da origen a nuevas variantes de los seres vivos, y como consecuencia, a nuevas especies.

En otras palabras, la evolución es un proceso de cambio que está guiado por la necesidad de adaptación ante cambios climáticos, geográficos y también guiado por la relación depredador-presa entre diferentes especies.

Los cambios adaptativos introducidos en una especie, acumulados a lo largo del tiempo, pueden dar origen a especies diferentes a aquellas en las cuales se produjeron las primeras variaciones.

Lo que Darwin plantea, es que aquellas características que permiten que los organismos tengan un mejor desempeño en la lucha por la adaptación, son seleccionados de manera natural para propagarse hacia las poblaciones descendientes.

Así visto, la evolución es un proceso que involucra descendencia y cambio. Este paso de privilegiar a los individuos mejor adaptados de una población para propagar sus características es lo que llamamos *selección natural*.

El concepto de mejor desempeño, o hablar de individuos mejor adaptados, depende del ambiente que estemos considerando. Las mismas características que permiten a un individuo *adaptarse* a un ambiente desértico y que le permiten sobrevivir, difieren mucho de las características que a un individuo de su misma especie, viviendo en el ártico, le permiten ese mismo nivel de supervivencia.

Esta idea es la que explotan los algoritmos genéticos, en el sentido que trabajan con poblaciones que también evolucionan, para una mejor *adaptación*, sólo que acá, la población no está constituida por seres vivos, sino que por *soluciones a un problema*.

Esto significa que tenemos una población de posibles soluciones a un problema dado y que a través de un *proceso evolutivo* se van modificando a través del tiempo de manera de tener soluciones más adaptadas a la solución de un problema particular.

2 Antecedentes históricos

Nuestro interés en el uso de técnicas evolutivas, está basado en la gran cantidad de problemas que se puede enfrentar con esta mirada. La búsqueda de soluciones de manera algorítmica se comporta de manera análoga a los aspectos señalados por la teoría de la evolución, en el sentido que se trata de operar sobre poblaciones de individuos que cambian sus características para alcanzar un mejor desempeño.

La idea de integrar la teoría de la evolución con sistemas de ingeniería no es nueva. En los años 50 George Box consideró variabilidad genética y selección natural, junto con un proceso basado en el concepto de *selección de los mejores*, en la operación en tiempo real de una planta de procesos [5].

En los años 60 en Alemania, ingenieros trabajaron en la aplicación de estrategias evolutivas para la búsqueda de soluciones en un espacio complejo que ofrecía 10^8 posibles soluciones, demostrando a través de experimentos lo adecuado que resultaba el uso de algoritmos evolutivos para enfrentar este tipo de problemas. A partir de ello, el uso de estrategias evolutivas tuvo un importante auge, especialmente en el campo de la optimización estructural.

Un investigador de la Universidad de Michigan llamado John Holland estaba consciente de la importancia de la selección natural, y a fines de los años 60 desarrolló una técnica para incorporarla en un programa de computador. Su objetivo era lograr que los computadores se comportaran como si tuvieran la capacidad de aprendizaje. A la técnica que inventó Holland se le llamó originalmente *planes reproductivos*, pero se hizo popular bajo el nombre *Algoritmo Genético* tras la publicación de su libro [2], aparecido en 1975.

Una definición bastante completa de un algoritmo genético es la que propone John Koza [3]:

Es un algoritmo matemático altamente paralelo que transforma un conjunto de objetos matemáticos individuales con respecto al tiempo usando operaciones modeladas de acuerdo al Principio Darwiniano de reproducción y supervivencia del más apto, y tras haberse presentado de forma natural una serie de operaciones genéticas de entre las que destaca la recombinación sexual. Cada uno de estos objetos matemáticos suele ser una cadena de caracteres (letras o números) de longitud fija que se ajusta al modelo de las



Figure 2: John Holland (1929-2015)

cadena de cromosomas, y se les asocia con una cierta función matemática que refleja su aptitud.

Una lectura esencial en el área de esta técnica evolutiva es el texto de Goldberg [1], que permite tener una buena percepción de las potencialidades de esta técnica. Del mismo modo, el texto de Melanie Mitchell [4] entrega una descripción detallada, a nivel introductorio, de los algoritmos genéticos.

3 ¿Cómo saber si es posible usar un algoritmo genético?

La aplicación m'as común de los algoritmos genéticos ha sido la solución de problemas de optimización, en donde han mostrado ser muy eficientes y confiables. Sin embargo, no todos los problemas resultan ser apropiados para la técnica en cuestión, y se recomienda, en general, tomar en cuenta las siguientes características del mismo antes de intentar su uso:

- Su espacio de búsqueda, i.e., sus posibles soluciones, debe estar delimitado dentro de un cierto rango.
- Debe ser posible definir una *función de aptitud* que nos indique qué tan buena o mala es una respuesta.

- Las soluciones deben poderse codificar de una forma que resulte relativamente fácil de implementar en un computador.

El primer punto es muy importante, y lo más recomendable es intentar resolver problemas que tengan espacios de búsqueda discretos - aunque éstos sean muy grandes. Sin embargo, también podrá intentarse usar la técnica con espacios de búsqueda continuos, pero preferentemente cuando existe un rango de soluciones relativamente pequeño.

La función de aptitud no es otra cosa que la función objetivo de nuestro problema de optimización. El algoritmo genético únicamente maximiza, pero la minimización puede realizarse fácilmente usando el recíproco de la función maximizante (debiendo cuidarse, por supuesto, que no se genere una división por cero). Una característica que debe tener esta función es que debe ser capaz de *penalizar* las malas soluciones, y *premiar* las buenas, de forma que sean estas últimas las que se propaguen con mayor facilidad.

La codificación más común de las soluciones suele darse a través de cadenas binarias, aunque la utilización de números reales y letras también resulta posible. El primero de los esquemas mencionados ha gozado de mucha popularidad debido a que forma parte de la proposición original de Holland, y fundamentalmente porque su implementación es muy sencilla.

4 Estructura básica de un algoritmo genético

Un algoritmo genético simple consiste de una población (inicial) de individuos, habitualmente generados de manera randómica, en la forma de una cadena de caracteres que se denominan *cromosomas*, en analogía al concepto de la evolución biológica. Los valores variables se representan generalmente en forma binaria, donde los valores asociados a las características representadas se denominan *genes* y los valores individuales (0 o 1) se denominan *alelos*. La figura 3 esquematiza estos conceptos.

El desempeño de un cromosoma, esto es, la valoración de cuán bueno es el elemento en términos de solución, se determina decodificando la representación binaria (el *genotipo*) y llevando el valor representado (el *fenotipo*) a un modelo matemático, llamado función de *fitness*. Esto permite valorar el cromosoma en función de sus características deseables para la solución de un determinado problema.

Cada problema tendrá por supuesto, su propia función de fitness.

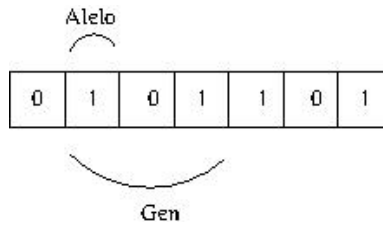


Figure 3: Esquema de un cromosoma

La generación siguiente, obtenida por la posible reproducción de los cromosomas presentes en la primera generación, dependerá del valor de fitness calculado para cada cromosoma.

Así, aquellos cromosomas con un alto valor de fitness tendrán una mayor probabilidad de ser elegidos para reproducción, mientras que aquellos que tienen un valor de fitness bajo, serán probablemente elegidos con mucha menos frecuencia para el proceso reproductivo.

El proceso de *Selección*, consiste entonces en la elección de aquellos candidatos a reproducirse, el que estará influenciado por el valor de fitness de cada cromosoma.

Además de la selección, hay otros dos operadores importantes para tratar con los cromosomas.

Uno de ellos se denomina *Cruzamiento* (cross-over), que permite que dos cromosomas seleccionados (llamados *padres*), aporten parte de su información (genética) para formar descendientes (los *hijos*) que tendrán una parte de sus genes heredadas de uno de los padres, y otra parte heredada del otro padre.

El otro operador interesante, llamado *Mutación*, permite modificar un gen en un cromosoma, aleatoriamente. Estos operadores serán explicados en detalle en apuntes siguientes.

El esquema general de un algoritmo genético se muestra en la figura 4.

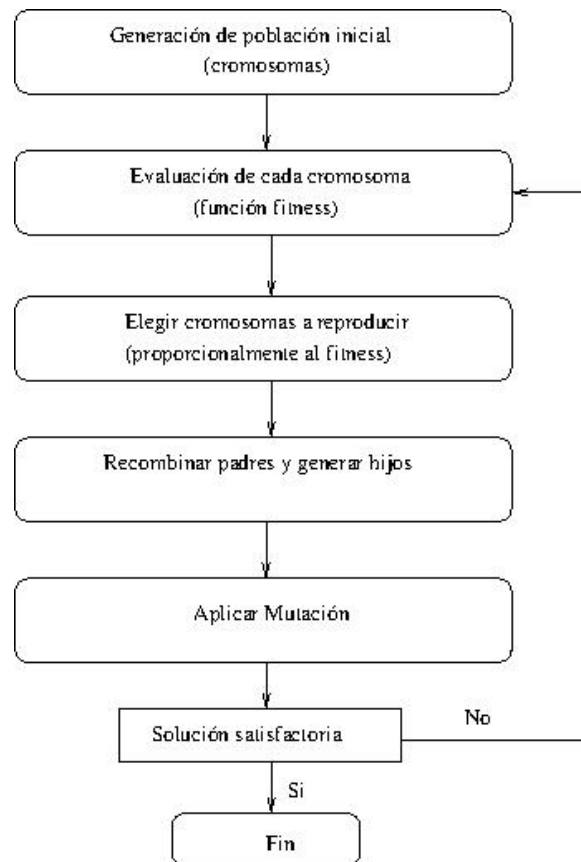


Figure 4: Esquema de la estructura fundamental de un algoritmo genético

References

- [1] Goldberg, David E. (1989). Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley Longman Publishing Co.
- [2] Holland, John (1975). Adaptation in Natural and Artificial Systems. The MIT Press.
- [3] Koza, J. R. (1992). Genetic Programming. On the Programming of Computers by Means of Natural Selection. The MIT Press.
- [4] Mitchell, Melanie (1998). An Introduction to Genetic Algorithms. Cambridge, Mass: MIT Press.

- [5] Parmee, Ian C. (2001). Evolutionary and Adaptative Computing in Engineering Design. Springer.